IEICE TRANS. INF. & SYST., VOL.E93–D, NO.1 JANUARY 2010

PAPER

Real-Time Estimation of Fast Egomotion with Feature Classification Using Compound Omnidirectional Vision Sensor

Trung Thanh NGO[†a)], **Yuichiro KOJIMA**[††], ***Nonmembers***, **Hajime NAGAHARA**[†††], **Ryusuke SAGAWA**[†], **Yasuhiro MUKAIGAWA**[†], **Masahiko YACHIDA**[††††], ***and*** **Yasushi YAGI**[†], ***Members***

SUMMARY For fast egomotion of a camera, computing feature correspondence and motion parameters by global search becomes highly time-consuming. Therefore, the complexity of the estimation needs to be reduced for real-time applications. In this paper, we propose a compound omnidirectional vision sensor and an algorithm for estimating its fast egomotion. The proposed sensor has both multi-baselines and a large field of view (FOV). Our method uses the multi-baseline stereo vision capability to classify feature points as near or far features. After the classification, we can estimate the camera rotation and translation separately by using random sample consensus (RANSAC) to reduce the computational complexity. The large FOV also improves the robustness since the translation and rotation are clearly distinguished. To date, there has been no work on combining multi-baseline stereo with large FOV characteristics for estimation, even though these characteristics are individually are important in improving egomotion estimation. Experiments showed that the proposed method is robust and produces reasonable accuracy in real time for fast motion of the sensor.

key words: *compound omnidirectional vision, multi-baseline stereo, large FOV, motion parameter separation, fast egomotion estimation, RANSAC*

1. Introduction

Egomotion, which consists of both rotation and translation, is an attractive research topic in computer vision and robotics. Using a camera is a common option in estimating egomotion. The egomotion of the camera is recovered by observing the motion on images. In realistic applications such as a wearable system, an unmanned aerial or land vehicle, fast motion usually occurs, especially with respect to rotation. Previous research work can be classified as using either a local search or global search for finding correspondences between consecutive frames and solving the egomotion estimation problem.

In the local search approach, a camera is assumed to move smoothly and slowly. Under this assumption, image feature points can be tracked for correspondence by a feature tracker [1] or an optical flow computation [2] through a video sequence. The camera egomotion is then estimated from the feature correspondence. However, the assumption is both too restrictive and ineffective in realistic applications such as aerial vehicles and wearable cameras where the motion is fast.

In the global search approach, random sample consensus (RANSAC) methods [3] solve the correspondence and camera motion estimation simultaneously. This approach searches globally for combinations that fit the motion hypotheses given by random sampling. Hence, there is no motion restriction. However, it is well-known that the computational cost of RANSAC increases exponentially according to the number of corresponding points. For 5-DOF egomotion estimation, we need five or seven feature correspondences between consecutive frames. Therefore, it is difficult to compute this problem in real-time using RANSAC methods.

Previous methods [4], [5] have separated the egomotion into rotation and translation to reduce the computational complexity and estimate them separately, a method which can be used for the global searching approach in real-time. If we separate the 5DOF egomotion into rotation and translation, we need at least two points of correspondence for the rotation estimation and two points of correspondence for the translation estimation. Hence, the computational cost of the estimation using RANSAC is drastically reduced and we should be able to apply it to real-time applications, even for unrestricted camera motion.

Utilizing sensor characteristics is one solution for separating motion or reducing computational complexity. It is well known that using viewpoints and FOV of visual sensors are two important ways to do this. Figure 1 shows the effect of these two characteristics for horizontal and vertical axes.

Stereo vision (as shown in the horizontal direction in Fig. 1) supplies depth information of feature points and thereby assisting motion matching using RANSAC methods to eliminate ambiguity [6], [7]. Depth information can also help us to separate egomotion into rotation and translation because the motion of far feature points on the image is mostly affected by the camera rotation. Hence, we can first estimate the rotation and then, by eliminating it, estimate the translation.

Large FOV sensors, such as omnidirectional vision sensors (as shown in the vertical direction in Fig. 1) have also frequently been used for motion estimation because the large FOV facilitates the observation of camera motion and

Manuscript received March 9, 2009.
Manuscript revised September 16, 2009.

[†]The authors are with the Institute of Scientific and Industrial Research, Osaka University, Ibaraki-shi, 567–0047 Japan.
[††]The author is with Sony Ericsson Mobile Communications, Tokyo, 108–0075 Japan.
[†††]The authors are with the Department of Systems Innovation, Graduate School of Engineering Science, Osaka University, Toyonaka-shi, 560–8531 Japan.
[††††]The author is with the Osaka Institute of Technology, Kitayama-shi, 573–0196 Japan.
a) E-mail: trung@am.sanken.osaka-u.ac.jp
DOI: 10.1587/transinf.E93.D.152

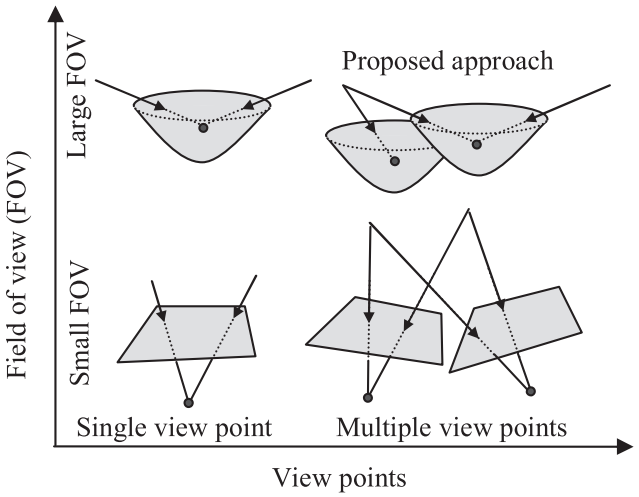


Fig. 1 Characteristics of optics for motion estimation.

improves the robustness of motion estimation. The motion flows caused by the camera rotation and translation do not always manifest themselves in a small FOV. For example, the focus of expansion (FOE) or focus of contraction (FOC) is usually out of view for traditional cameras, causing the egomotion algorithm to be sensitive to the orientation of the camera. By contrast, an omnidirectional camera always captures the FOE and FOC in its large FOV. When a traditional camera with a small FOV rotates around its vertical axis or translates parallel to the image plane, these motions produce quite similar motion flows on the image. However, the motion flows on the image are quite distinct for large FOV images.

Hence, multiple viewpoints and a large FOV are important factors for egomotion estimation. Nevertheless, there has been no previous research that combines both multiple views and large FOV characteristics (the diagonal direction shown in Fig. 1) for egomotion estimation of a sensor.

Therefore, in this paper, we propose a new stereo omnidirectional vision sensor, which has multi-baseline stereo and a large FOV for focusing on egomotion estimation. We also propose a real-time egomotion estimation algorithm for the compound omnidirectional vision sensor. Instead of using accurate depth information of feature points, the proposed method classifies the image features into far and near features according to the characteristics of small baseline stereo. Then we estimate the camera rotation and translation separately using the far and near features, respectively. The aim of the proposed method is to reduce the computational cost and improve the robustness of the estimation. When compared with conventional methods using an omnidirectional camera, the proposed method has an advantage in the real-time estimation of fast camera motion since the motion constraint is not applicable. The compound omnidirectional sensor also has some advantages compared with a conventional stereo system. First, it is small and lightweight making it suitable for wearable systems. Second, the binary classification of near and far feature points is significantly faster when checking a precomputed scene-independent look-up table. Finally, stereo matching of feature correspondences between mirror images is not required.

The rest of this paper is organized as follows. Section 2 discusses the related work. Section 3 provides an overview of our proposed algorithm. Section 4 describes the compound sensor and feature classification. Section 5 describes rotation and translation estimations and their optimization using RANSAC. We evaluate our experiments in Sect. 6, and this is followed by the conclusions.

2. Related Works

2.1 Sensors

In recent related works on the sensor, the stereo omnidirectional sensor is preferred in computer vision and robotics. The difference between sensors lies in how simple the calibration is and how the stereo disparity varies depending on the direction of the sensor. Some sensors consist of two cameras [25], [26] with parabolic or hyperbolic mirrors. However, they have only one baseline and the stereo sensitivity is not omnidirectional. Using two cameras means that the photometry of both cameras must be calibrated. In other works [27], [28] have used a catadioptric stereo system with a single camera and multiple mirrors to generate multiple virtual viewpoints. They compute depth by epipolar geometry between the virtual cameras. The photometry calibration of these sensors is not necessary. However, the FOV and the number of baselines are limited, therefore the stereo sensitivity is not omnidirectional. For this reason, a sensor has been developed [29], [30] consisting of a single camera and multiple spherical mirrors, which has multiple baselines, so that the stereo sensitivity is omnidirectional. This omnidirectional stereo system supplies the full omnidirectional capability that can detect near objects efficiently. However the sensor does not have a single center of projection (the projective model of a pin-hole camera) for each stereo view, and ignores the viewpoint differences along the vertical FOV. We used paraboloidal mirrors and an orthographic camera for a compound sensor in this study. We could achieve accuracy with the single center of projection for each view and the stereo views between the mirrors, which made it easier to detect the feature distance without any assumptions.

2.2 Egomotion Estimation

There are many methods [8]–[10] for simultaneous localization and map building (SLAM). SLAM methods simultaneously estimate sensor egomotion and build an environmental map. However, there is a fundamental difference between SLAM and egomotion methods. SLAM requires lengthy observation to obtain an estimation. When tracking frequently fails such as in the case of fast and sudden egomotion, SLAM methods are not effective for estimating egomotion. On the other hand, the proposed egomotion estimation can be applied with fast and sudden camera motion

and does not require lengthy observation to initialize and obtain the estimation. Hence, SLAM is not considered as related work here.

Most algorithms estimate rotation and translation simultaneously, several algorithms separate rotation and translation and estimate them separately to reduce the computational complexity. The egomotion separation methods can be classified into several groups: fixation methods, translation-limited, rotation-limited and depth-based methods.

Fixation methods [4] simplify the egomotion estimation by holding the observation direction to the same environmental point through time. Therefore, fixation reduces the number of unknowns from five to four. Egomotion is divided into rotation and translation and estimated separately. However, keeping track of the same environmental point through a video sequence is quite difficult due to problems such as occlusion and noise. Moreover, a minimum of four point correspondences for estimating egomotion is still time-consuming in the case of global searching with RANSAC.

In rotation-limited methods, the assumption of small rotation is required to approximate a rotation matrix and it can be represented by a vector of rotation angles, which helps to decouple the rotation and translation. Then some algorithms are used [12], [13] to estimate the first translation by computing the FOE and then estimating the rotation. Other algorithms [11] estimate rotation first and then translation. However, these algorithms still require five point correspondences for the estimation, which is time-consuming in the case of global searching with RANSAC.

In the translation-limited methods, a lot of research assumes the motion is only rotation (3DOF) and the computation is fast. The rotation matrix is estimated by the matching of features on consecutive images [14], [15] or using least squares of points correspondences [16], [17]. In these papers, the authors estimate only rotation, however, further efforts [18], [19] can be made to estimate camera translation after eliminating camera rotation of the feature points. In these approaches, we may need only two point correspondences for estimating rotation and two point correspondences for estimating translation.

Our method belongs to the group of egomotion methods that use depth information. Previous methods belonging to this group [20]–[23] use depth information to improve the robustness of the estimation and tracking of feature correspondence, and not for reducing the computational complexity. Our proposed method uses the depth information to separate egomotion into rotation and translation to reduce the computational cost. However, instead of using an accurate depth, we use the binary depth information by classifying feature points into near and far features. Rotation is estimated using only far features and translation is estimated using only near features. The dimensions of the data point and the dimensions of motion parameters are reduced. Two point correspondences are needed for estimating both rotation and translation.

3. Overview of the Proposed Algorithm

The proposed algorithm estimates the camera egomotion with 5 degrees of freedom, 3D rotation and 2D translation (the direction of translation without magnitude). The flowchart for the algorithm is presented in Fig. 2. We used a compound omnidirectional sensor, described in the next section. The sensor provided a compound image that consists of all mirror's omnidirectional images. Corner feature points are detected in the center omnidirectional image, and are then classified as near or far features. We estimate the egomotion from two successive video frames. Rotation is estimated using only the far features, while translation is estimated using only the near features after eliminating the rotational motion using the estimated rotation. Since we are dealing with large camera motion, tracking image features is not helpful. Therefore, we use the RANSAC search [31] to find the global correspondences and to estimate the egomotion simultaneously. During this process, we could estimate rotation and translation separately to reduce the computation complexity, because we have already classified the feature as either far or near. Finally, rotation and translation parameters are optimized under epipolar constraints using all the supporters from the RANSAC estimation.

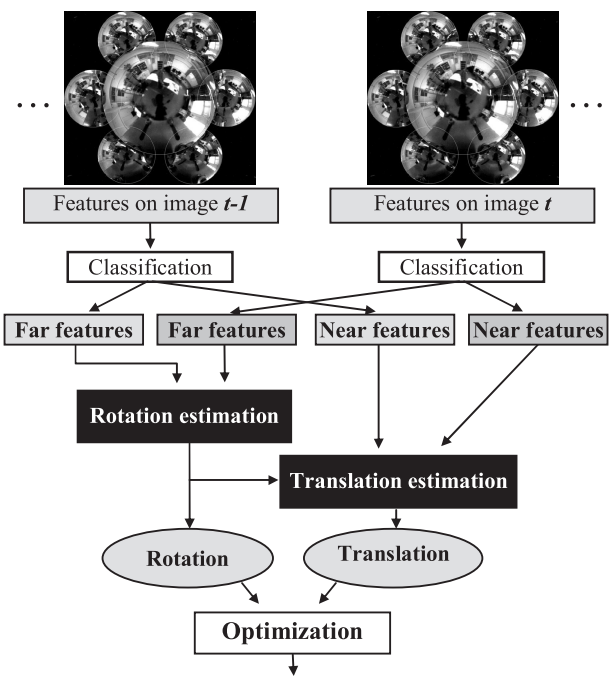


Fig. 2 Algorithm flowchart: Detected features are classified as near or far features; rotation is estimated using the far features while translation is estimated using the near features; finally, optimization is performed to tune the estimated motion parameters.

4. Compound Omnidirectional Vision Sensor and Feature Classification

4.1 Compound Omnidirectional Vision Sensor

A compound omnidirectional vision sensor consists of M paraboloidal mirrors, one large mirror at the center with $M-1$ small surrounding mirrors, and a single camera. In a 3D coordinate system, each mirror i has the parameter (r_i, O_i), where r_i is its radius of curvature at the top and O_i is its location. The baseline between a pair of mirrors (i, j) is defined as $b_{i,j} = \sqrt{\|O_i - O_j\|}$. Figure 3 shows one example of our compound sensor, in which $M = 7$ paraboloidal mirrors and a single orthographic camera are used. Mirror i, $i = 1..7$, has the diameter d_i, and the total diameter of the compound mirror is D.

A light ray from an object is projected onto the image plane by reflection from the mirrors. Since the position of each mirror is different, the distance of an object can be computed by triangulation. However, the baseline of triangulation is very narrow since it is the distance of the reflected points on the mirrors. Hence, it is not practical to use this sensor to compute with accuracy the distance of an object. Instead, we classify objects into two categories, near and far objects.

For a mirror i, $i = 1, \ldots, M$, the mirror coordinate system originates at the optical center, the vertical axis z_i points towards its top along the symmetrical axis of the mirror, and the whole camera coordinate system coincides with the coordinate system of the center mirror, see Fig. 7. The shape of the surface of a paraboloidal mirror is described as the function of the spherical coordinate system:

$$\tau = \frac{r_i}{1 + \cos(\phi)}, \tag{1}$$

where r_i is the radius of curvature of the mirror i at its top, (τ, ϕ, θ) is a point on the surface of the paraboloidal mirror i in the spherical coordinate system, τ is the distance from the origin to the surface point, $0 \le \phi \le \pi$, $-\pi \le \theta \le \pi$. The top of the mirror has the coordinates $(\tau, \phi, \theta) = (\frac{r_i}{2}, 0, 0)$. An object point P from the ray direction (ϕ, θ), which is represented by $\boldsymbol{P} = (\sin(\phi)\cos(\theta), \sin(\phi)\sin(\theta), \cos(\phi))$, has the projection on the image plane with the coordinates:

$$\boldsymbol{x_i} = \left\{ O_i^x + \frac{r_i \sin(\phi)\cos(\theta)}{1 + \cos(\phi)}, O_i^y + \frac{r_i \sin(\phi)\sin(\theta)}{1 + \cos(\phi)} \right\}, \tag{2}$$

where (O_i^x, O_i^y) is the center of an omnidirectional image from the mirror i, measured in pixel units is the result of the calibration process. Since we use the orthographic image sensor, the geometric parameter calibration of the paraboloidal mirrors is simple and has been reported in previous work [32]. Therefore it is not shown in this paper.

Since the total size of the constructed compound mirror is less than 50 mm, the stereo baseline is very narrow, which means that the resolution of the computing distance is low. Therefore, we propose classifying the distance of the image features as either near or far, instead of computing an accurate distance. We have found this method to be useful and a small system is sufficient for estimating egomotion.

4.2 Classification of Near and Far Features

In this section, we describe the classification for an object point on one pair of mirrors, say mirrors i and j, then describe the classification for multiple pairs of mirrors, since the sensor consist of seven mirrors.

We consider the situation in which an object is placed at an infinite distance from the sensor and observed in the images of two mirrors. As the object gets close to the sensor along the ray of one of the mirrors, the projected image on the other mirror shifts along the epipolar line. This shift is called disparity. In this paper, we consider an object to be far if the disparity is less than a given threshold.

Figure 4 shows an example of the epipolar line for the paraboloidal mirrors i and j. Their center points in the image plane are $\boldsymbol{O_i}$ and $\boldsymbol{O_j}$, respectively. If an object is at an infinite distance, the ray direction is $\boldsymbol{P}$. The rays are projected on $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ after being reflected, at $\boldsymbol{m_i}$ and $\boldsymbol{m_j}$ on the mirrors.

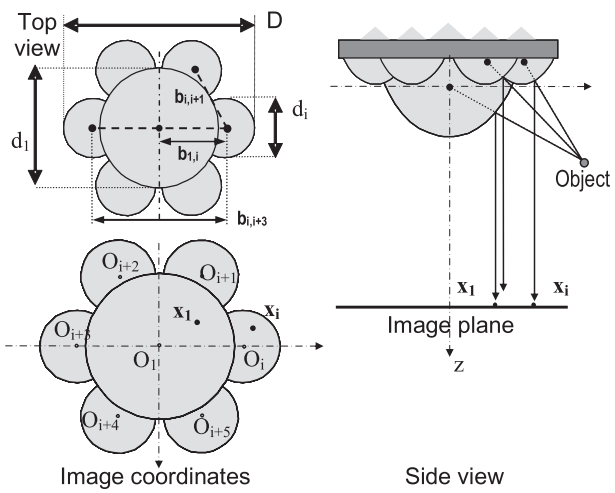


Fig. 3 An example of a compound omnidirectional sensor: a single orthographic camera with seven compound paraboloidal mirrors.

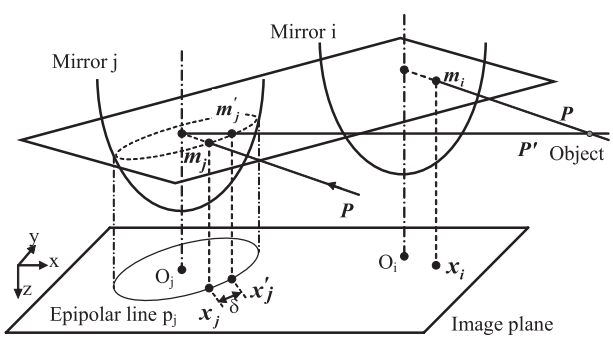


Fig. 4 The ray directions reflected on one pair of compound paraboloidal mirrors (i and j) and the epipolar line.

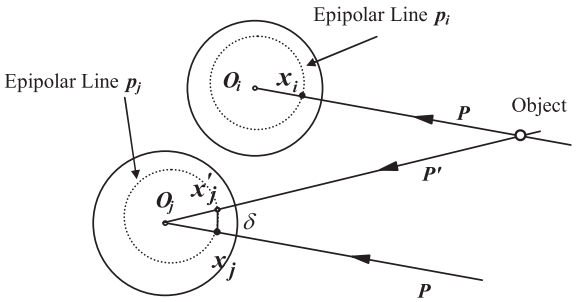


Fig. 5 Projection of ray directions and epipolar lines on the image plane for one pair of compound mirror i and j.

If the object moves closer to mirror i along the direction $\boldsymbol{P}$, the ray direction to mirror j is $\boldsymbol{P}'$. Thus, the reflected and projected points become $\boldsymbol{m_j}'$ and $\boldsymbol{x_j}'$. $\boldsymbol{x_j}'$ is shifted δ pixels along the epipolar line. There are two such epipolar lines p_i and p_j for i and j, respectively. Figure 5 shows more detail of the projection on the image plane.

Since the proposed method is a narrow baseline stereo, the disparity δ is small if an object is at a practical distance. Therefore, we compute the disparity without searching corresponding points thus enabling real-time computation. Since we apply this method to feature points detected by a feature detector, we assume that the intensity around $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ along their epipolar lines are step functions defined as:

$$\begin{cases} I_i(p) = H(p - p_{\boldsymbol{x_i}}) \\ I_j(p) = H(p - p_{\boldsymbol{x_j}} - \delta), \end{cases} \tag{3}$$

where $H(x)$ is a step function, that is 1 if $x \geq 0$ and 0 otherwise, p indicates the position in the epipolar line and $p_{\boldsymbol{x_i}}$, $p_{\boldsymbol{x_j}}$ are the position of $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ in their epipolar lines, respectively.

The Lucas–Kanade method [33] computes disparity from the gradient of intensity without searching for correspondences. However, the gradient-based method cannot be applied directly to the case assumed in (3). Therefore, we filter the images before computing disparity. Our method smoothes the intensity along the epipolar line using a 1D mean filter. Figure 6 shows an example of this. The thin lines are the original intensities of two images along the epipolar lines. The black and gray lines denote images i and j. The shift between the black and gray lines represents the disparity. After applying a mean filter of window size $2n + 1$ to $I_i(p)$ and $I_j(p)$, we obtain the smoothed functions indicated by the thick black and gray lines. Then, we can compute the disparity from the gradient of the smoothed functions as follows:

$$D_{i,j}(\boldsymbol{P}, n) = \frac{I_i(p_{\boldsymbol{x_i}}) - I_j(p_{\boldsymbol{x_j}})}{I_i(p_{\boldsymbol{x_i}} + n) - I_i(p_{\boldsymbol{x_i}})}. \tag{4}$$

If the disparity δ is less than n, $D_{i,j}(\boldsymbol{P}, n) < 1$, otherwise $D_{i,j}(\boldsymbol{P}, n) \geq 1$. Therefore, we classify a feature point in the direction $\boldsymbol{P}$ according to the following criterion:

$$\begin{cases} \text{Far feature} & \text{if} \quad D_{i,j}(\boldsymbol{P}, n) < 1 \\ \text{Near feature} & \text{otherwise.} \end{cases} \tag{5}$$

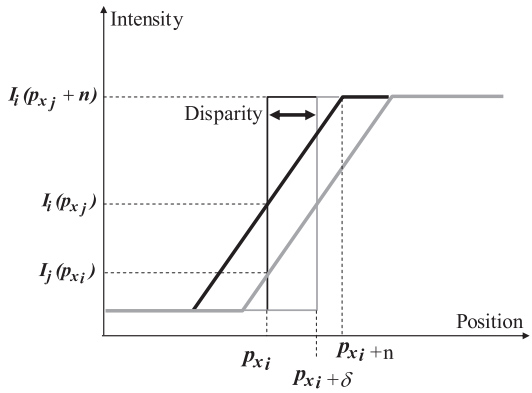


Fig. 6 The disparity δ is computed using a gradient-based method after smoothing with a mean filter. Two intensity functions along the epipolar lines p_i and p_j are aligned at $p_{\boldsymbol{x_i}}$ and $p_{\boldsymbol{x_j}}$, $p_{\boldsymbol{x_i}} = p_{\boldsymbol{x_j}}$.

The classification criterion is adjusted by the window size of the mean filter n. If a user wants to discriminate features at a certain distance, the disparity d corresponding to the distance can be computed from the optical geometry of the sensor. Thus, the classification is achieved by setting $n = d$. Since d differs with respect to the position in the image of our sensor, we compute d that corresponds to the distance of discrimination for each pixel of the image. In implementation, all the positions in (4) can be computed offline and stored in a look-up table. Then in run-time, the disparity can be checked quickly.

Since our sensor has seven mirrors, it can compute disparity by using different pairs (i, j) of mirrors. If a feature is observed by multiple pairs, it is determined to be a near one if it is classified as near by more than one of these pairs.

5. Estimation of Rotation and Translation with RANSAC

The proposed method estimates rotation and translation separately using features classified as either near or far features. The main contribution is that the method reduces the computational complexity by estimating rotation and translation separately, instead of estimating them together. Moreover, since the correspondences of features between consecutive video frames are also determined using a RANSAC-based algorithm, the proposed method can be applied to fast egomotion.

In the following sections, we describe separating the estimation of rotation and translation using the classified features, explain the RANSAC-based algorithm to find the correspondences and to estimate motion simultaneously and describe optimization to refine the estimated motion parameters.

5.1 Separate Estimation of Rotation and Translation

Once the coordinate system has been located at the optical center O of the center mirror as shown in Fig. 7, the surrounding scene moves around the sensor. If the position of

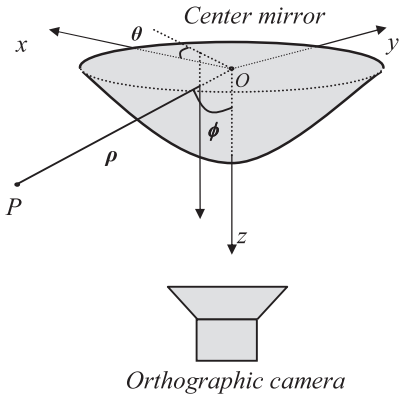


Fig. 7 Camera coordinate system with origin at the center of central mirror.

object point P is $\rho \boldsymbol{P}$, where ρ is the distance from P to the origin and $\boldsymbol{P} = (\sin\phi\cos\theta, \sin\phi\sin\theta, \cos\phi)$ is the direction from the origin to P. The position $\rho' \boldsymbol{P'}$ after rigid transformation is given by:

$$\rho' \boldsymbol{P'} = \boldsymbol{R}\rho\boldsymbol{P} + \boldsymbol{T}, \tag{6}$$

where $\boldsymbol{P}$, $\boldsymbol{P'}$ are coordinates on the unit sphere or the directions of P before and after the motion, ρ and ρ' are its depths before and after the motion, $\boldsymbol{R}$ is the rotation matrix and $\boldsymbol{T}$ the translation vector.

It is noted that the depth, ρ and ρ', are not known from the captured image, while $\boldsymbol{P}$ and $\boldsymbol{P'}$ are known. By subdividing (6) by ρ', we obtain

$$\boldsymbol{P'} = \frac{\rho}{\rho'}\boldsymbol{R}\boldsymbol{P} + \frac{\boldsymbol{T}}{\rho'}. \tag{7}$$

From (7) we see that if the distance ρ' is much larger than $\boldsymbol{T}$, we can ignore the term $\frac{\boldsymbol{T}}{\rho'}$. Therefore, if P is a far feature, the motion is determined only by rotation and $\frac{\rho}{\rho'} \approx 1$. Equation (7) is then simplified as follows:

$$\boldsymbol{P'} = \boldsymbol{R}\boldsymbol{P}. \tag{8}$$

Consequently, we can estimate rotation $\boldsymbol{R}$ separately by omitting the translation $\boldsymbol{T}$ from the equation. Translation is then estimated after eliminating the estimated rotation from the motion.

5.2 Computing Rotation Independently

In conventional methods, at least five pairs of corresponding feature points between two images are required to estimate rotation and translation. But since we can distinguish far feature points, we can compute rotation separately. We can thus reduce the required number of feature points to two pairs.

In Fig. 8, $\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{P'}$ and $\boldsymbol{Q'}$ are the projected points of the two points P and Q before and after a rotation, respectively. If we consider the cross-product vector $\boldsymbol{n}$ of $\boldsymbol{P}$ and $\boldsymbol{Q}$, the cross-product vector $\boldsymbol{n'}$ of $\boldsymbol{P'}$ and $\boldsymbol{Q'}$ is given by applying

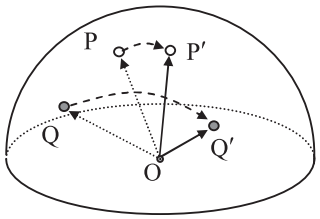


Fig. 8 Estimate rotation from motion of 2 points.

the same rotation to $\boldsymbol{n}$ as follows:

$$\boldsymbol{P'} = \boldsymbol{R}\boldsymbol{P} \quad \boldsymbol{Q'} = \boldsymbol{R}\boldsymbol{Q} \quad \boldsymbol{n'} = \boldsymbol{R}\boldsymbol{n}, \tag{9}$$

where the lengths of $\boldsymbol{n}$ and $\boldsymbol{n'}$ are normalized to the unit length. Then the rotation matrix $\boldsymbol{R}$ is computed from the three pairs of vectors on the unit sphere:

$$\boldsymbol{R} = [\boldsymbol{P'}\ \boldsymbol{Q'}\ \boldsymbol{n'}]\,[\boldsymbol{P}\ \boldsymbol{Q}\ \boldsymbol{n}]^{-1}, \tag{10}$$

and $\boldsymbol{R}$ is normalized, $|\boldsymbol{R}| = 1$. In the estimation algorithm using RANSAC, this computation is used to initialize the rotation model, which needs only two points of correspondence in two consecutive frames.

Having finished the random sampling using the RANSAC method, the best rotation matrix $\boldsymbol{R}_{est}$ and a set S_R of k feature correspondence pairs between two video frames are outputted:

$$\boldsymbol{P'}^i = \boldsymbol{R}_{est}\boldsymbol{P}^i, \tag{11}$$

where $i = 1..k$, $k > 3$. We then solve the over-determined equation system (11) for three rotation angles. This equation system can be solved using the least mean squares method (LMS) with the minimization function:

$$\min_{\phi,\theta,\psi} \sum_{\boldsymbol{P}\in S_R} (\boldsymbol{P'} - \boldsymbol{R}(\phi,\theta,\psi)\boldsymbol{P})(\boldsymbol{P'} - \boldsymbol{R}(\phi,\theta,\psi)\boldsymbol{P})^T, \tag{12}$$

where $\boldsymbol{R}(\theta,\phi,\psi)$ is a rotation matrix built from three angles θ, ϕ, ψ. Since the minimization is nonlinear, we used the Levenberg-Marquardt minimization with the initial parameters given by $(0,0,0)$. After the minimization, the output rotation matrix, $\boldsymbol{R}(\theta_c,\phi_c,\psi_c)$, clearly satisfies the conditions of a rotation matrix, the orthogonality condition and its determinant being +1.

5.3 Computing Translation after Eliminating Rotation

Once rotation has been estimated, we can eliminate the rotation of features. Therefore, in this section we assume that no rotation occurs between two succeeding views. The translation vector can now be estimated from the motion of two near feature points.

Consider a situation in which the camera moves while observing two near feature points P and Q as shown in Fig. 9. These points are projected, onto $\boldsymbol{P}$ and $\boldsymbol{Q}$ in the previous video frame, and $\boldsymbol{P'}$ and $\boldsymbol{Q'}$ in the current video frame. Now, we consider two planes, π_P and π_Q. π_P is created by

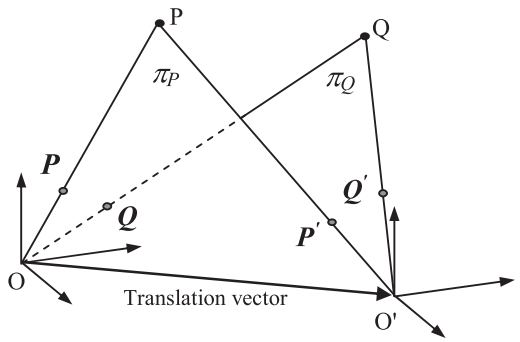


Fig. 9 O and O' are located on the intersection of two epipolar planes.

three points, O, O', and P. Similarly, π_Q is created by O, O', and Q. Since the translation vector $\boldsymbol{T}$ is the intersection of the two planes, $\boldsymbol{T}$ is computed as follows:

$$\boldsymbol{T} = (\boldsymbol{P} \times \boldsymbol{P'}) \times (\boldsymbol{Q} \times \boldsymbol{Q'}). \tag{13}$$

Since the motion of the feature points in the image occurs as a result of the translation of the camera, the projection of the translation vector and the motion vector of the features on the image plane must be opposite. We use this criterion to adjust the direction of the translation vector.

5.4 RANSAC-Based Methods to Estimate Rotation and Translation

The RANSAC-based algorithms are implemented similarly for both rotation and translation estimation. For both algorithms, a random sample is taken of two features in the previous frame and two more in the current frame. The algorithms simultaneously find the motion parameters and the correspondence of image features. The difference is the feature used for estimation. For rotation estimation, near features which do not hold the rotation represented by (8) are filtered out by our compound sensor. Since it is not feasible to use far features for estimating the translation, these should be excluded from the translation estimation and only near feature points should be used for this task.

The RANSAC estimation of both rotation and translation is summarized as follows:

1. Randomly select two features in the first video frame and two image features in the second video frame to assign two pairs of correspondence.
2. Calculate the motion parameters (rotation matrix $\boldsymbol{R}$, or translation vector $\boldsymbol{T}$).
3. Count the supporting pairs of correspondence that match the estimated parameters.
4. Record the current best solution with the maximum number of supporting pairs.
5. If not stopped, return to the first step.

In our implementation, the termination criterion for RANSAC sampling is processing time.

For estimating rotation, the rotation matrix $\boldsymbol{R}$ is computed as shown in Sect. 5.2. Counting supporting pairs for the rotation $\boldsymbol{R}$ is done by applying the rotation of far features in the previous frame and matching these features to the features in the current frame. If $\boldsymbol{P}$ is a far feature in the previous frame and $\hat{\boldsymbol{P}}$ is the position after applying the estimated rotation, we compute the angle between $\hat{\boldsymbol{P}}$ and all far features in the current frame located near $\hat{\boldsymbol{P}}$. If the angle between $\hat{\boldsymbol{P}}$ and $\boldsymbol{P'}$ is the smallest and less than a threshold, we count $(\boldsymbol{P}, \boldsymbol{P'})$ as a supporting pair of correspondence.

For estimating translation, the translation vector $\boldsymbol{T}$ is computed as shown in Sect. 5.3 for each random sample. Then the number of supporting pairs for each translation vector is counted to select the best translation vector. From a near feature point $\boldsymbol{P}$ in the previous frame and $\boldsymbol{T}$, an epipolar plane $\pi(\boldsymbol{P}, \boldsymbol{T})$ can be computed. Similarly, an epipolar plane $\pi(\boldsymbol{P'}, \boldsymbol{T})$ can be computed for each near feature $\boldsymbol{P'}$ of the current frame. If the angle between the normal vectors of $\pi(\boldsymbol{P}, \boldsymbol{T})$ and $\pi(\boldsymbol{P'}, \boldsymbol{T})$ is the smallest and less than a threshold, we count $(\boldsymbol{P}, \boldsymbol{P'})$ as a supporting pair of correspondence. Therefore, counting the supporting pairs for the translation vector is a problem of stereo matching.

5.5 Optimizing the Solution Using Epipolar Constraint and All Supporting Pairs

After the estimation using the RANSAC-based method, the approximate rotation matrix $\boldsymbol{R}(\phi_c, \theta_c, \psi_c)$ and translation vector $\boldsymbol{T}_{est}$ are acquired. An optimization is performed to find the best rotation angles and translation vector using all the supporting pairs that have been obtained from the RANSAC-based method. The optimized parameters are estimated by minimizing the following epipolar constraint function:

$$\min_{\theta,\phi,\psi,tx,ty,tz} \sum_{(\boldsymbol{P},\boldsymbol{P'})\in S} (\boldsymbol{P'} E \boldsymbol{P})^2, \tag{14}$$

where S is the set of all pairs of feature points that support the best solution estimated using the RANSAC-based method, and $(\boldsymbol{P},\boldsymbol{P'})$ is one of these pairs. E is the essential matrix computed as $E = [\boldsymbol{T}]_\times \boldsymbol{R}(\theta, \phi, \psi)$, where $\boldsymbol{R}(\theta, \phi, \psi)$ is the rotation matrix built from the rotation angles (θ, ϕ, ψ), and $[\boldsymbol{T}]_\times$ is the matrix representation of the cross product with $\boldsymbol{T} = (t_x, t_y, t_z)$; see [38] for further details. We also used the Levenberg-Marquardt minimization with the initial parameters given by $\boldsymbol{R}(\theta_c, \phi_c, \psi_c)$ and $\boldsymbol{T}_{est}$. After the optimization, the rotation matrix is $\boldsymbol{R}(\theta_{opt}, \phi_{opt}, \psi_{opt})$, which obviously meets the conditions of a rotation matrix.

5.6 Computational Cost of Estimation Using RANSAC

In the proposed approach, the correspondence and motion are determined in a RANSAC procedure. This approach cannot be applied to the well-known seven-point algorithm, because the computational cost is prohibitive. Assuming a problem that has no prior knowledge about feature correspondences, we formulate the computational cost of our proposed algorithm compared with that of the seven-point

algorithm which estimates the essential matrix without feature correspondence. For a RANSAC algorithm, the number of iterations required before the estimation obtains a correct sample is:

$$k = \frac{\log z}{\log(1 - w^n)}, \tag{15}$$

where z is the probability of seeing only bad samples, w is the fraction of inliers among all data points and n is the number of data points for one sample; refer to the book [34] for the details. Let p_{in} be the probability of selecting an inlier $\boldsymbol{P}$ in the previous frame, and p_{sup} be the probability of selecting a correct supporter $\boldsymbol{P'}$ in the current frame. $\boldsymbol{P'}$ can be found in the region around the location of $\boldsymbol{P}$ and the region contains about m features (in the current frame), then $p_{sup} = \frac{1}{m}$. The value of p_{sup} is similar for both the proposed algorithm and the seven-point algorithm, however the value of p_{in} differs. In our proposed method, the inliers are far features at a certain distance from the sensor, and far features are classified by the compound sensor. Therefore, not all far features classified by the sensor are inliers. Thus the average value of p_{in} in the proposed algorithm is smaller than that in the seven-point algorithm. The probability of selecting the correct correspondence pair $(\boldsymbol{P}, \boldsymbol{P'})$ is $w = p_{in}p_{sup}$. For the proposed algorithm, the probability of selecting two pairs of feature correspondence is $w_2 = (p_{2in}p_{2sup})^2$. For the seven-point algorithm the probability of selecting seven pairs of feature correspondence is $w_7 = (p_{7in}p_{7sup})^7$.

To ensure the same possibility of obtaining a correct sample, the value of z must be equal for both algorithms, therefore, the number of required iterations must vary to meet the requirement. We have simulated how these two algorithms require the number of iterations in the case that $p_{7in} = 1.5p_{2in}$ and $z = 0.9$. Details of the required number of iterations are described in Fig. 10, which shows that the seven-point algorithm requires many more iterations compared with the proposed algorithm. For example, if $m = 10$, there are about 10 features in the current frame around the location of an inlier in the previous frame, and therefore $p_{sup} = 0.1$, and $k_7 = 294042$. This can be compared with the proposed algorithm giving $k_2 = 16.4$ only. From these analyses, the proposed method drastically decreases the computation cost owing to the separation of the camera motion.

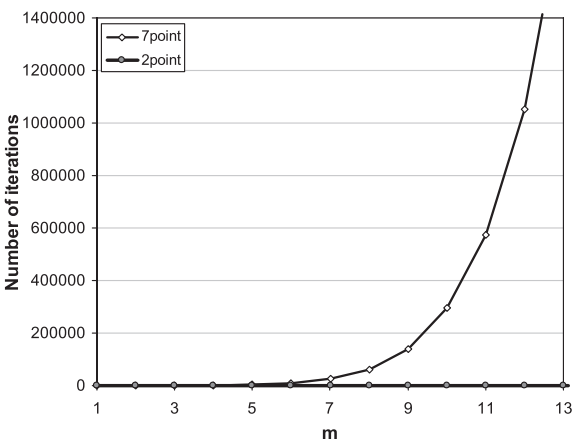


Fig. 10 Number of required iterations for the seven-point algorithm without feature correspondence and the proposed algorithm.

6. Experiments

In our experiments, we used the compound omnidirectional sensor that is shown in Fig. 3. The compound sensor is mounted on a system with two rotary stages and a 50 cm translation stage (Fig. 11). The ω_θ rotation is controlled by one rotation controller on the z-axis, and the ω_ϕ rotation on the y-axis is controlled by the other. The dimensional translation of the camera system is controlled by a translation controller. The vision sensor is a 1600×1200 pixel CCD camera (Scorpion: Point Grey Research) with a telecentric lens. The parameters of the compound sensor and its parameters after the calibration are shown in Table 1. In the experiments, the maximum distance for classification by this compound sensor is about 3 m. The proposed method is processed by a PC with a Pentium D 3.2 GHz processor. OpenCV [36] is used for image processing including the Harris [35] feature detection procedure.

The experiments were carried out in various envi-

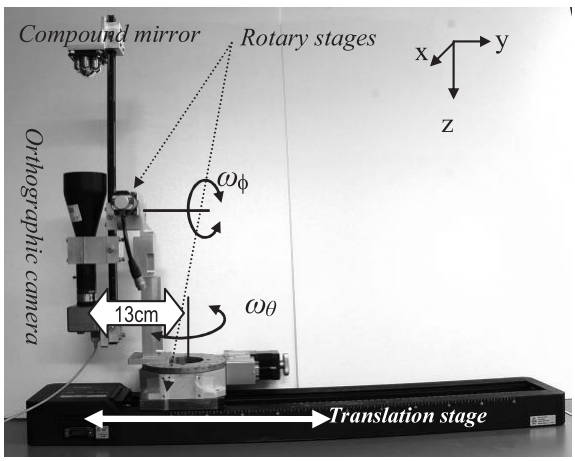


Fig. 11 The evaluation system. Rotary stages and the vision sensor are mounted on the translation stage.

Table 1 Parameters of the compound sensor after calibration.

Parameter		Design (mm)	Actual size in image (pixel)
Total diameter	D	43	813
Center mirror radius	d_1	25	473
Side mirror radius	d_i	13	239
Center mirror radius of curvature	r_1	17.5	331
Side mirror radius of curvature	r_i	8.7	165
Largest baseline (side mirrors)	$b_{i,i+3}$	30	567
Smallest baseline (side mirrors)	$b_{i,i+1}$	15	283
Side mirror - center mirror baseline	$b_{1,i}$	19.5	369

ronments to evaluate accuracy with respect to processing time and camera motion. Our experimental results were compared with the results from the essential matrix-based solution. We implemented the seven-point algorithm based on the work of Torr [37] that estimates the essential matrix using RANSAC and the multi-resolution Kanade-Lucas-Tomasi (KLT) feature tracker [1] implemented in OpenCV [36]. Motion parameters are also tuned by using the same optimization method as our proposed algorithm. In this method, which we refer to as 7ALGRANSAC, the feature correspondences are given by the feature tracker including outlier correspondences. While it is possible to implement the seven-point algorithm using RANSAC without knowing the correspondences, it is very time-consuming to sample a set of 7 correspondences between two consecutive frames. Consequently, we do not cover this implementation in the paper. We also compared the performance of the proposed algorithm with and without feature classification using the compound sensor to show the effectiveness of the near/far feature classification procedure. The detailed results of these experiments are described in the following sections, which show the averages of the frame-by-frame estimation errors for each video sequence.

Several types of environmental data were captured in our experiments to validate our algorithm. We extracted 200 features from each frame using a Harris feature detector. The whole sensor image is used for feature classification; and the big omnidirectional image at the center is used for feature detection. The experiment showed that for each frame the Harris feature detector needed 0.066 sec to extract 200 features.

We also set up the parameters so that our algorithm could cope with a maximum rotation velocity of 31 degrees/frame and a translation velocity of 8.5 cm/frame. For our algorithm, the processing time includes feature extraction, feature classification and RANSAC motion estimation, and Levenberg Marquardt optimization of the motion parameters. By contrast the processing time for the 7ALGRANSAC process includes initial feature detection, frame by frame feature tracking, RANSAC estimation of the essential matrix, motion parameter extraction and optimization using the Levenberg Marquardt method.

6.1 Error Definitions

Errors of motion are defined for motion between a pair of video frames. To evaluate the rotation error, we first compute the residual rotation after eliminating the estimated motion $\hat{\boldsymbol{R}}$ with the true motion $\boldsymbol{R}_{tr}$ from the rotary stage controller:

$$\boldsymbol{R}_{er} = \hat{\boldsymbol{R}}\boldsymbol{R}_{tr}^{-1} \tag{16}$$

This is the error of the estimated rotation that is represented by a matrix. If the estimation is perfect, the matrix $\boldsymbol{R}_{er}$ is the identity rotation matrix. The difference between $\boldsymbol{R}_{er}$ and the identity rotation matrix $\boldsymbol{I}$ is assumed to be the error of estimation. We evaluate the rotation error by a Frobenius norm of the matrix $(\boldsymbol{R}_{er} - \boldsymbol{I})$ as follows:

$$\sqrt{\sum_{i=1,j=1}^{3,3} (\boldsymbol{R}_{er,ij} - \boldsymbol{I}_{ij})^2}. \tag{17}$$

If the error is small, it can be regarded as the angle error in radians.

The translation error is the angular difference between the normalized estimated translation vector and the normalized ground-truth translation vector, because our method estimates translation without magnitude. We call this the directional translation error, which is also measured in radians.

6.2 Experiments with Different Ratios of Near/Far Features

First experimental data were captured for three different ratios of near/far features. These data sets are labelled *FAR*, *MID*, *NEAR*, and are shown in Fig. 12, with a decreasing number of far features (or an increasing number of near features). Near features were situated within 3 m of the sensor, whereas far features were located at distances ranging from 4 m to about 10 m. Motion of the sensor was controlled by only the rotary stage ω_θ on the Oz axis. While the sensor was rotated, it was also translated. The motion path was circular with a radius of 13 cm.

For these data sets, experimental results were obtained for feature classification, the convergence of rotation and translation estimation.

6.2.1 Feature Classification

First, we tested the accuracy of classification using the proposed sensor. We manually checked the classified results with the ground-truth and summarized the results of feature classification using 10 random frames. For the ground-truth, a feature is classified as near if the distance is less than 3 m; otherwise it is classified as far. Table 2 gives a summary of feature classification for the three data sets, which shows that the accuracy of feature classification is more than 90%.

6.2.2 Convergence of RANSAC for Estimating Rotation

Next, we evaluate the accuracy of rotation estimation with respect to processing time. The processing time includes the Harris feature detection with and without feature classification and the RANSAC matching time for estimating rotation. The camera translation and rotation angles were fixed as the control rotation velocity $\omega_\theta = 10$ degrees/frame. We compared the convergence of estimating rotation with feature classification (denoted as CLASSIFIED) and without feature classification (denoted as UNCLASSIFIED). The results are shown in Fig. 13.

Because most outliers for estimating rotation were removed by classification, the processing time was reduced significantly for the CLASSIFIED case compared with that

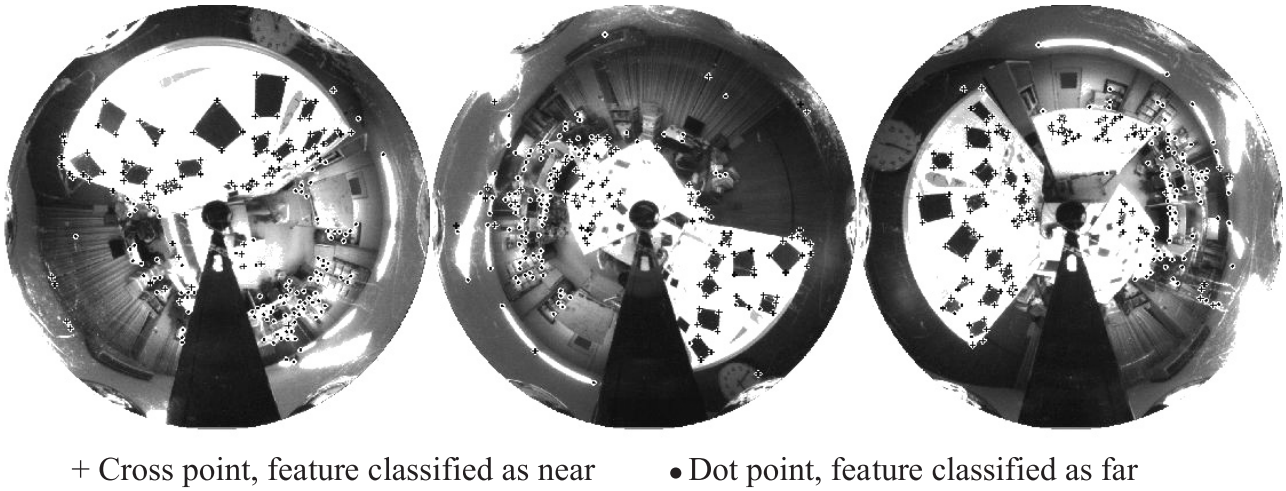


Fig. 12 Example input images (each of them is the big omnidirectional image at center of a sensor image) of *FAR*, *MID*, *NEAR* (from left to right) with increasing near/far ratios.

Table 2 Results of misclassification of near/far features.

	FAR	*MID*	*NEAR*
Near→ Far	4.4%	8.4%	9.8%
Far→ Near	2.1%	9.0%	5.3%
Actual number of near features	92	105	112
Actual number of far features	108	95	88

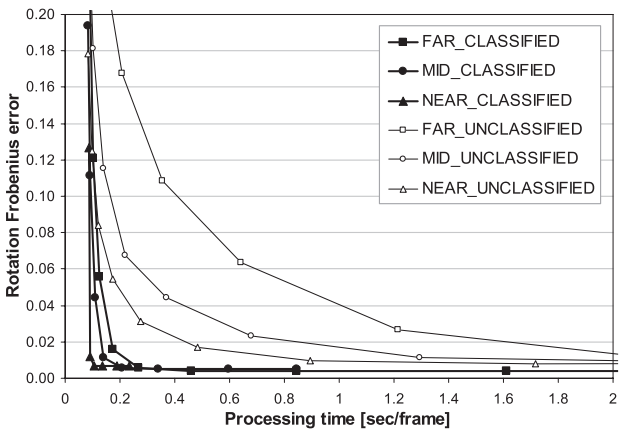


Fig. 13 Comparison of the convergence of estimating rotation with/ without feature classification.

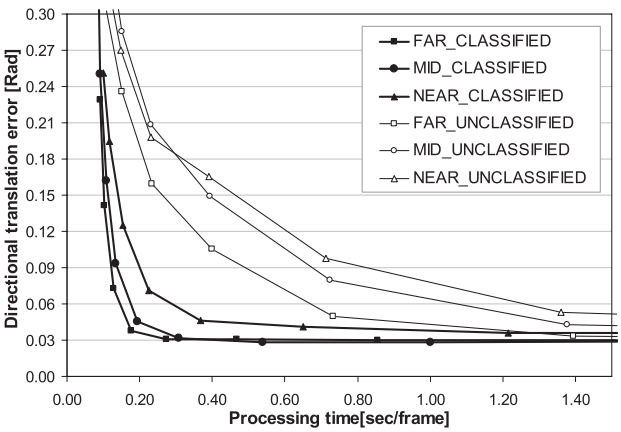


Fig. 14 Comparison of the convergence of translation estimation with/ without feature classification using ground-truth rotation.

for the UNCLASSIFIED case. The processing time of 0.2 sec is reasonable for use in real applications with acceptable accuracy. Since the far features were not truly at infinity and the rotation matrix is computed from four random points on two images and no optimization was performed after random sampling, some error existed in the estimation regardless of the processing time.

6.2.3 Convergence of RANSAC for Estimating Translation

The experiments for the convergence of translation estimation were carried out with the same rotation estimation, which in this case was ground-truth rotation. The camera translation and rotation velocities were fixed with the control rotation velocity $\omega_\theta = 10$ degrees/frame. The processing time consisted of the Harris feature detection with/without feature classification and the RANSAC matching time for estimating translation. Figure 14 shows the results of convergence for both classified and unclassified features. The results show that the translation estimation with only near features converged much faster than in the unclassified case. The results are similar to those in Fig. 13 showing that the classification of features is effective. Since the translation vector is computed from four random points on two images and no optimization was performed after random sampling, some error existed in the estimation regardless of the processing time.

From the experiments on the convergence of rotation and translation estimation using classified and unclassified features, we can see that with classification of features, the egomotion (rotation and translation) computation is much faster than is the case without feature classification but with

+ Cross point, feature classified as near
• Dot point, feature classified as far

Fig. 15 Indoor scene 1.

+ Cross point, feature classified as near
● Dot point, feature classified as far

Fig. 16 Indoor scene 2.

the same processing time.

6.3 Overall Performance Experiments

In these experiments, the performance of the proposed algorithm was tested with various real data. Two indoor video sequences and one outdoor scene were captured, as shown in Figs. 15, 16 and 23, respectively. For these videos, the sensor was moved by both a translation stage controller and a rotation stage controller instead of using only a rotation stage controller as in the previous data sets. The speed of the translation stage was fixed at 5 cm/frame, while the rotation speed ω_θ varied between 12 and 30 degrees/frame; one video was taken at each rotation speed. In the experiments, the processing time allowed for each algorithm was 0.5 sec/frame for terminating the RANSAC iteration.

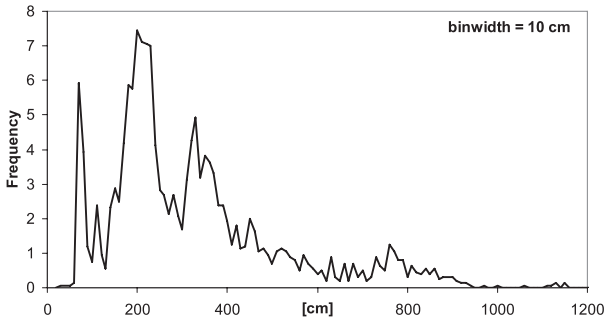


Fig. 17 Indoor scene 1: histogram of feature distances.

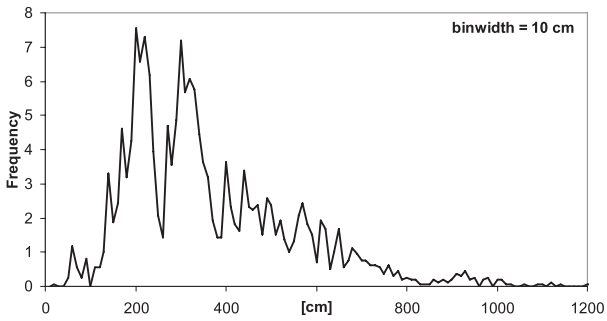


Fig. 18 Indoor scene 2: histogram of feature distances.

6.3.1 Indoor Scenes

Examples of feature classification in the two scenes are shown in Figs. 15 and 16, while the distributions of the distances from the sensor to the feature points are shown in Figs. 17 and 18 for the two scenes, respectively. The distance distributions are presented as the distance histograms with the bin-width of 10 cm. The distributions of the far feature points for these two scenes are similar, whereas the distributions of near feature points differ. The near feature points in the first scene are closer to the sensor. The distances were computed using stereo matching for the central omnidirectional images and the baseline connecting two ends of the translation stage, the length of which is 50 cm. A series of captures provided us the average distribution as shown in Figs. 17 and 18.

The experimental results are described in Figs. 19 and 20 for the first indoor scene and Figs. 21 and 22 for the second indoor scene. For smaller motion, the performance of 7ALGRANSAC is better than the proposed algorithm; however for larger motion, the proposed algorithm gives the better results. Since 7ALGRANSAC relies on the feature correspondences from a feature tracker, the estimation accuracy

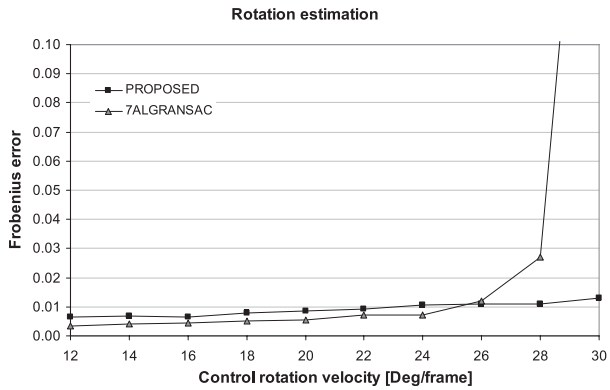


Fig. 19 Indoor scene 1: frobenius error for rotation estimation.

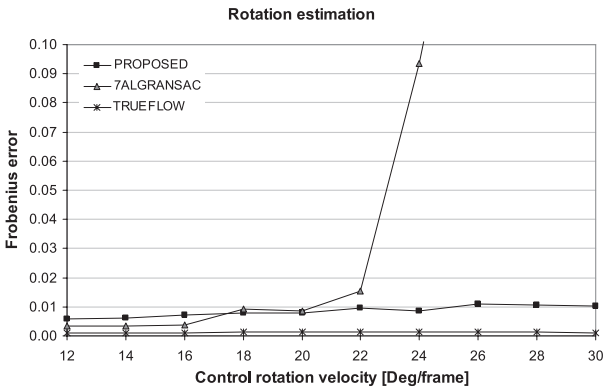


Fig. 21 Indoor scene 2: frobenius error for rotation estimation.

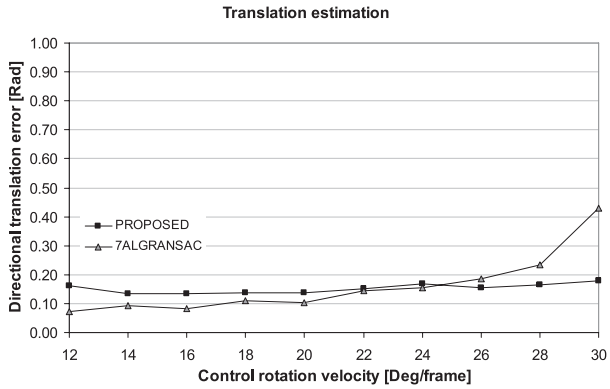


Fig. 20 Indoor scene 1: directional translation error for translation estimation.

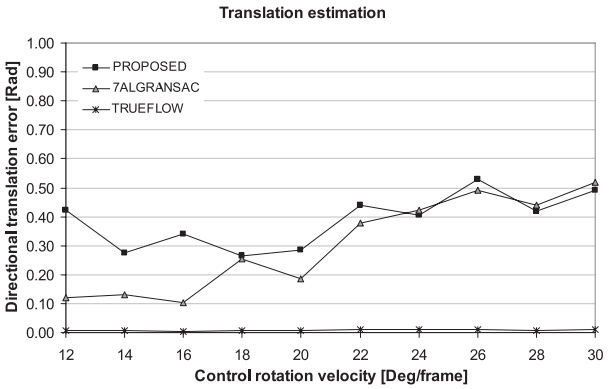


Fig. 22 Indoor scene 2: directional translation error for translation estimation.

of 7ALGRANSAC decreases with greater motion, as many outliers are included in the correspondences. Since our algorithm does not rely on correspondences from a feature tracker, the performance is robust for any amount of motion. Rotation estimation is a little less accurate with greater motion. This can be explained by the larger translation of the sensor, because our algorithm assumes that the distance to far features is much larger than the translation speed of the sensor.

We can also see that with the same camera motion, the translation estimation in the first scene in Fig. 20 is better than that in Fig. 22, while the rotation estimation accuracy is similar in two scenes. The reason for the difference is that the near feature points in the first indoor scene are distributed closer to the sensor, while the distributions of the far feature points are similar. A similar variation in the accuracy of 7ALGRANSAC was observed for these scenes.

6.3.2 Outdoor Scene

In a outdoor scene, far feature points are significantly farther from the sensor than those in the indoor scenes, and there are fewer near feature points than in either of the indoor scene. One of the outdoor scenes is shown in Fig. 23 and the distribution of feature distances from the sensor is shown in Fig. 24 as a distance histogram with a bin-width of 10 cm. For this scene, most feature points are very far from the sensor, with few near feature points located around the sensor and on the ground. The experimental results are described in Figs. 25 and 26. The results are similar to those of the indoor scenes. For slow motion, the proposed algorithm and 7ALGRANSAC produced similar results; however the proposed algorithm gave better results for fast motion. The proposed algorithm produced robust results for all variations in motion speed.

We can also see that for the outdoor scene, the far feature points are farther away, the approximation error is lower, and we have a more accurate rotation estimate. And since there are very few near feature points, the translation estimation accuracy for both algorithms is not as good as that in the indoor scenes. Due to the few near feature points in this scene, 7ALGRANSAC did not work well. 7ALGRANSAC simultaneously estimates rotation and translation, and therefore if translation is not accurate, rotation is directly affected.

6.4 Discussion

The proposed algorithm relies on separate camera motion estimations. The rotation is estimated using far feature points while the translation is estimated using near feature points. Far and near feature points are classified us-

Fig. 23 Outdoor scene.

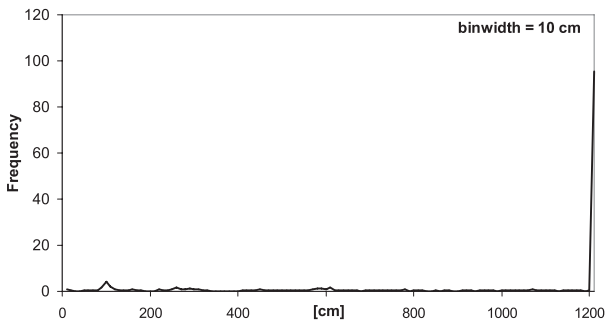


Fig. 24 Outdoor scene: histogram of feature distances.

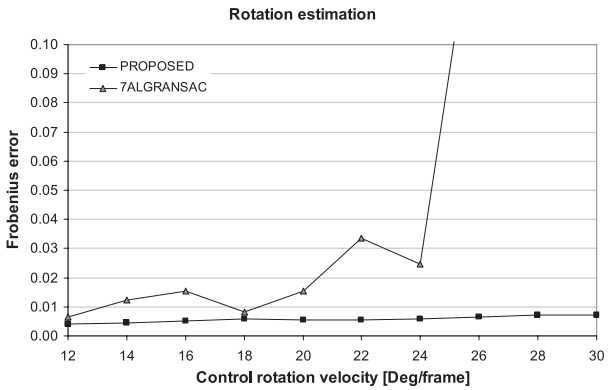


Fig. 25 Outdoor scene: frobenius error for rotation estimation.

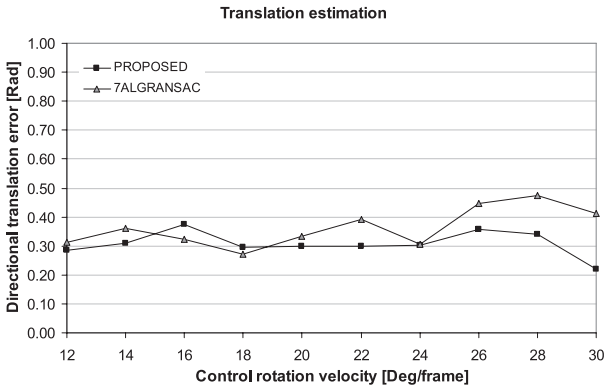


Fig. 26 Outdoor scene: directional translation error for translation estimation.

ing the proposed compound omnidirectional sensor. There are some exceptional cases in which the proposed algorithm does not work well, but these are not seen as a disadvantage of the proposed method. The first situation arises when the scene is small and all feature points are classified as near features, with the results that we have no far features for estimating camera rotation. However, for fast and sudden camera motion in a small environment, the previous egomotion algorithms also have problems in computing the feature correspondence. This needs to be addressed in our future research. The second situation arises when all the feature points are very far from the sensor and are classified as far features. In this situation, the rotation can be accurately estimated by our algorithm, however, we have no near features for estimating camera translation. Previous egomotion algorithms also face the same problem because the translation vector is relatively too small to be estimated effectively. The algorithms that simultaneously estimate rotation and translation do not work well in this situation because an inaccurate translation estimation directly influences the rotation accuracy. However, algorithms that separate rotation and translation, such as the proposed algorithm, work better.

The distances of feature points can affect the accuracy of the estimation. For translation estimation, this influence is well-known for previous algorithms and the proposed algorithm. If feature points are relatively closer to the sensor, then higher accuracy of translation we can get and otherwise. However, for rotation estimation, we approximate the rotation by the motion of far feature points. The farther the distances of far features, the higher accuracy we have.

In the current algorithm, we only use the geometry constraint for computing the camera motion. Obviously, if we can apply the similarity constraint of feature points the results can be significantly improved. Some feature descriptor such as SIFT [39] can be used in such a situation. We can also improve the RANSAC estimation by applying an adaptive-threshold robust estimator [40], which is an improvement of RANSAC that does not require the user-defined threshold. In this robust estimator, the threshold to separate inliers is adaptively estimated depending on the distribution of the residuals of data points.

The current algorithm works well without motion blur or with only a small amount of blur. However, we need to improve the algorithm to work with the motion blur caused by faster motion.

7. Conclusion

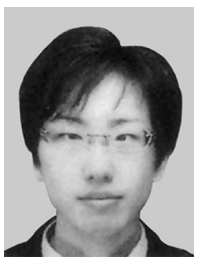
In this paper, we have proposed a new compound omnidirectional sensor and a method for estimating egomotion which applies the RANSAC process. Using the proposed sensor, image features are classified into near or far features. The rotation of the camera is estimated using only the far features, since the motion of far features in the images is modeled solely by rotation. After estimating the rotation, the translation is estimated using only the near features. Therefore, only two pairs of features are required to estimate either rotation or translation, whereas the seven-point algorithm requires seven pairs of features. Because of this reduction in computational complexity, the proposed method can work in real time without being given correspondences. Consequently, it can compute large camera motion since it does not assume small motion is required to find correspondences by a conventional feature tracker.

References

[1] J. Shi and C. Tomasi, "Good features to track," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.593–600, 1994.
[2] B. Horn and B. Schunck, "Determining optical flow," Artif. Intell., vol.17, pp.185–204, 1981.
[3] D. Nister, "Preemptive RANSAC for live structure and motion estimation," Ninth IEEE International Conference on Computer Vision (ICCV'03) - vol.1, pp.199–206, 2003.
[4] K. Daniilidis, "Fixation simplifies 3D motion estimation," Computer Vision and Image Understanding, vol.68, no.2, pp.158–169, 1997.
[5] S. Tsuji, Y. Yagi, and M. Asada, "Dynamic scene analysis for a mobile robot in a man-made environment," Proc. IEEE Robotics and Automation Conf., vol.2, pp.850–855, 1985.
[6] C.F. Olson, L.H. Matthies, M. Schoppers, and M.W. Maimone, "Stereo ego-motion improvements for robust rover navigation," Proc. IEEE Robotics and Automation Conf., vol.2, pp.1099–1104, 2001.
[7] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," Proc. IROS, pp.3946–3952, 2008.
[8] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces, mixed and augmented reality," 6th IEEE and ACM International Symposium on, pp.225–234, 2007.
[9] A.J. Davison, "Real-time simultaneous localization and mapping with a single camera," Proc. ICCV, vol.2, pp.1403–1410, 2003.
[10] E. Mouragnon, F. Dekeyser, P. Sayd, M. Lhuillier, and M. Dhome, "Real time localization and 3D reconstruction," Proc. CVPR, vol.1, pp.363–370, 2006.
[11] K. Prazdny, "Egomotion and relative depth map from optical flow," Biol. Cybernatics, vol.36, pp.87–102, 1980.
[12] D.J. Heeger and A.D. Jepson, "Subspace methods for recovering rigid motion I: Algorithm and implementation," Int. J. Comput. Vis., vol.7, no.2, pp.95–117, 1992.
[13] H.C. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," Proc. R. Soc. London Ser. B 208, pp.385–397, 1980.
[14] M.E. Albert and J.H. Connell, "Visual rotation detection and estimation for mobile robot navigation," Proc. IEEE Robotics and Automation Conf., vol.5, pp.4247–4252, 2004.
[15] M. Brown and D.G. Lowe, "Recognising panoramas," Ninth IEEE International Conference on Computer Vision (ICCV'03), vol.2, pp.1218–1225, 2003.
[16] N. Ohta and K. Kanatani, "Optimal estimation of three-dimensional rotation and reliability evaluation," ECCV 1998, pp.175–187, 1998.
[17] S.Z. Li, H. Wang, and W.Y.C. Soh, "Robust estimation of rotation angles from image sequences using the annealing M-estimator," Journal of Mathematical Imaging and Vision, vol.8, no.2, pp.181–192, 1998.
[18] T. Okatani and K. Deguchi, "Estimating camera translation based on a voting method using a camera with a 3D orientation sensor," Proc. ICPR, vol.1, pp.275–278, 2002.
[19] Z.Z. Chen, N. Pears, J. McDermid, and T. Heseltine, "Epipole estimation under pure camera translation," Sun, C., Talbot, H., Ourselin, S., Adriaansen, T. (Eds.), Proc. Seventh Digital Image Computing, CSIRO, Collingwood, Sydney, pp.849–858, 2003.
[20] S. Negahdaripour, N. Kolagani, and B. Hayashi, "Direct motion stereo for passive navigation," Proc. Conf. Computer Vision and Pattern Recognition, pp.425–431, June 1992.
[21] G.P. Stein and A. Shashua, "Direct estimation of motion and extended scene structure from a moving stereo rig," Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.211–218, June 1998.
[22] G.S. Young and R. Chellappa, "3-D motion estimation using a sequence of noisy stereo images: Models, estimation, and uniqueness results," PAMI(12), no.8, pp.735–759, Aug. 1990.
[23] T.S. Huang and S.D. Blostein, "Robust algorithms for motion estimation based on two sequential stereo image pairs," CVPR, pp.518–523, 1985.
[24] K. Tan, H. Hua, and N. Ahuja, "Multiview panoramic cameras using mirror pyramids," IEEE Trans. Pattern Anal. Mach. Intell., vol.26, no.7, pp.941–946, 2004.
[25] J. Gluckman and S. Nayar, "Real-time omnidirectional and panoramic stereo," Proc. Image Understanding Workshop, pp.299–303, 1998.
[26] A. Chaen, K. Yamazawa, N. Yokoya, and H. Takemura, "Omnidirectional stereo vision using hyperomni vision," IEICE Technical Report, pp.96–122, Feb. 1997.
[27] J. Gluckman and S. Nayar, "Rectified catadioptric stereo sensors," IEEE Trans. Pattern Anal. Mach. Intell., vol.24, no.2, pp.224–236, 2002.
[28] J. Baldwin and A. Basu, "3D estimation using panoramic stereo," Proc. IAPR Int. Conf. on Pattern Recognition, pp.1097–1100, 2000.
[29] R. Sagawa, N. Kurita, T. Echigo, and Y. Yagi, "Compound catadioptric stereo sensor for omnidirectional object detection," Proc. IEEE/RSJ IROS, vol.2, pp.2612–2617, 2004.
[30] Y. Kojima, R. Sagawa, T. Echigo, and Y. Yagi, "Calibration and performance evaluation of omnidirectional sensor with compound spherical mirrors," The 6th Workshop on Omnidirectional Vision, Camera Networks and Non-classical cameras, 2005.
[31] M.A. Fischler and R.C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," Comm. of the ACM 24, pp.381–395, June 1981.
[32] R. Sagawa, N. Aoki, and Y. Yagi, "Mirror localization for catadioptric imaging system by observing parallel light pairs," Proc. 8th Asian Conference on Computer Vision, LNCS 4843, pp.116–126, Tokyo, Japan, Nov. 18-22, 2007.
[33] B.D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," IJCAI81, pp.674–679, 1981.
[34] D.A. Forsyth and J. Ponce, Computer vision: A modern approach, Prentice Hall, 2002.
[35] C. Harris and M.J. Stephens, "A combined corner and edge detector," Proc. 4th Alvey Vision Conference, pp.147–151, 1988.
[36] Intel Corporation, Open Source Computer Vision Library.
[37] P.H.S. Torr, "Outlier detection and motion segmentation," PhD dissertation, Dept. of Eng. Science, Univ. of Oxford, 1995.
[38] H.C.L. Higgins, "A computer algorithm for reconstructing a scene from two projections," Nature 293, pp.133–135, 1981.

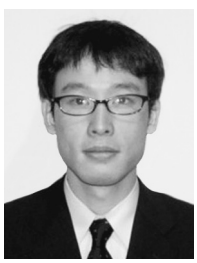
[39] D.G. Lowe, "Object recognition from local scale-invariant features," Proc. Seventh IEEE International Conference on Computer Vision, vol.2, pp.1150–1157, Sept. 1999.
[40] T.T. Ngo, H. Nagahara, R. Sagawa, Y. Mukaigawa, M. Yachida, and Y. Yagi, "An adaptive-scale robust estimator for motion estimation," Proc. 2009 IEEE International Conference on Robotics and Automation, pp.2455–2460, May 12-17, 2009.

Trung Thanh Ngo received his M.E. degree from Osaka University in 2005. He is currently a Ph.D. candidate at Osaka University and working as a Research Assistant in the Department of Intelligent Media, Institute of Scientific and Industrial Research, Osaka University.



Yuichiro Kojima received his M.E. degree from Osaka University in 2007. He is currently working for Sony Ericsson Mobile Communications.

Hajime Nagahara received B.E. and M.E. degrees in Electrical and Electronic Engineering from Yamaguchi University in 1996 and 1998, respectively. He received his Ph.D. in System Engineering from Osaka University in 2001. He was a Research Associate of the Japan Society for the Promotion of Science in 2001-2003. He was a Research Associate of Graduate School of Engineering Science, Osaka University, in 2003-2006. He was a Visiting Associate Professor at CREA University of Picardie Jules Verns, France in 2005. He has been an Assistant Professor of Graduate School of Engineering Science since 2007. He was a Visiting Researcher at Columbia University, USA in 2007-2008. His research interests include image processing, computer vision and virtual reality. He received an ACM VRST2003 Honorable Mention Award in 2003.



Ryusuke Sagawa is an Assistant Professor at the Institute of Scientific and Industrial Research, Osaka University, Osaka, Japan. He received a BE in Information Science from Kyoto University, Kyoto, Japan, in 1998. He received an M.E. in Information Engineering in 2000 and a Ph.D. in Information and Communication Engineering from the University of Tokyo, boxTokyo, Japan in 2003. His primary research interests are computer vision, computer graphics and robotics (mainly geometrical modeling and visualization). He is a member of IPSJ, RSJ, and IEEE.

Yasuhiro Mukaigawa received his M.E. and Ph.D. degrees from The University of Tsukuba in 1994 and 1997, respectively. He became a research associate at Okayama University in 1997, an assistant professor at University of Tsukuba in 2003, and an associate professor at Osaka University in 2004. His current research interests include photometric analysis and computational photography. He was awarded the MIRU Nagao Award in 2008. He is a member of IPS, VRSJ, and IEEE.

Masahiko Yachida received the B.E., and M.S. degrees in Electrical Engineering, and a Ph.D. in Control Engineering, all from Osaka University, Osaka, Japan in 1969, 1971, and 1976, respectively. He joined the Dept. of Control Engineering, Faculty of Engineering Science, Osaka University in 1971 as a Research Associate and became an Associate Professor at the same department. He then moved to the Dept. of Information and Computer Science as Professor in 1990 and was a Professor at the Dept. of System Engineering at the same University. He was a Professor of Systems and Human Science, Graduate School of Engineering Science, Osaka University. He has beed a Professor in the Faculty of Information Science and Technology, Osaka Institute of Technology since 2008. He is an author of Robot Vision (Shoko-do, received Ohkawa Publishing Prize), a co-author of Pattern Information Processing (Ohm-sha) and an editor of Computer Vision (Maruzen) and other books. He was Chairman of Technical Committee on Computer Vision, Information Processing Society of Japan and a Chairman of Technical Committee on Pattern Recognition & Media Understanding, Institute of Electronics Information & Communication Engineers, Japan. His interests are in the field of computer vision, image processing, mobile robotics and artificial intelligence.

Yasushi Yagi is the Professor of Intelligent Media and Computer Science, and the Assistant Director of the Institute of Scientific and Industrial Research, Osaka University, Ibaraki, Japan. He received his Ph.D. degree from Osaka University in 1991. In 1985, he joined the Product Development Laboratory, Mitsubishi Electric Corporation, where he worked on robotics and inspections. He became a Research Associate in 1990, a Lecturer in 1993, an Associate Professor in 1996, and a Professor in 2003 at Osaka University. International conferences for which he served as chair include: FG1998 (Financial chair), OMINVIS2003 (Organizing chair), ROBIO2006 (Program co-chair), ACCV2007 (Program chair), PSVIT2009 (Financial chair) and ACCV2009 (General chair). He is the Editor of IEEE ICRA Conference Editorial Board (2007, 2008), the Editor-in-chief of IPSJ Transactions on Computer Vision & Image Media, and the Associate Editor-in-chief of IPSJ Transactions on Computer Vision & Applications. He was awarded the ACM VRST2003 Honorable Mention Award, IEEE ROBIO2006 Finalist of T.J. Tan Best Paper in Robotics, IEEE ICRA2008 Finalist for Best Vision Paper, and MIRU2008 Nagao Award. His research interests are computer vision, medical engineering and robotics. He is a member of IPS, RSJ, and IEEE.